View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by UUM Repository

Journal of ICT, 13, 2014, pp: 125–144

TECHNICAL CONSIDERATIONS IN DESIGNING HAPTIC APPLICATIONS: A CASE STUDY OF LAPAROSCOPY SURGERY SIMULATION WITH HAPTIC ELEMENTS

[1]Norkhairani Abdul Rawi, [2]Halimah Badioze Zaman & [3]Azlina Ahmad

[1]Faculty of Informatics, Universiti Sultan Zainal Abidin, Malaysia
[2, 3]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia

khairani@unisza.edu.my; hbz@ftsm.ukm.my; aa@ftsm.ukm.my

ABSTRACT

Simulation has been widely used as a tool for training, especially in high risk areas such as in the aerospace, military and medical fields. Surgery is one of the sub areas that has been receiving much attention from researchers due to the ability of simulation to provide a real surgery setting and human organs with appropriate devices to increase the realism. Developing a surgical simulation is a technically complex process since it involves a few components that interact with each other. Thus this necessitates further considerations regarding the issues and challenges in order to produce an accurate and interactive application. Therefore there is a need for a technical solution framework to help a new and novice researcher in this area to get started. This paper discusses the important components of a surgical simulator, together with its issues and challenges. A proposed solution framework, together with the programming or application choices that are available for each of the components, is explained clearly as concluded from the discussion in the previous work. The class structure for the components is shown briefly to give the new researcher an idea of it. It is hoped that this paper will serve as a foundation for new and novice researchers in haptic development specifically and visual informatics generally.

Keywords: Surgical simulation, haptic applications, technical framework, visual informatics.

INTRODUCTION

In most areas, highly skilled employees are really needed to reduce failure and error. Skills are normally acquired through the teaching of experts together with consistent practice (Wong & Matsumoto, 2008; Hamdorf & Hall, 2000). The more time spent between the experts and the trainees the more will be the increase in skills acquisition in preparing a new batch of experts. In the medical field, especially in surgery, researches are being carried out to reduce the incisions in surgery, thus leading to the implementation of MAS (minimal access surgery) or what is known as laparoscopy. This procedure needs to be performed by highly skilled surgeons, and it is quite difficult for the surgical trainee to master this skill (Wong & Matsumoto, 2008; Hamdorf & Hall, 2000). Therefore, there is a need for the surgical trainee to spend more time with the experts in order to acquire this skill under expert supervision. Anyway, this has given rise to some issues such as the limitation of time, as well as social and financial constraints that prevent the experts from spending more time in training the trainees (Hamdorf & Hall, 2000; Norkhairani, Halimah, & Azlina, 2011; Najmaldin, 2007).

All the above issues have then led to the use of technology in an effort to help the trainee to do frequent and consistent training. Robotics, simulations and virtual training systems are the technologies that are available to fill this gap (Panait et al., 2009). Besides that, the use of dummies also helps in the acquisition of surgical skills (Panait et al., 2009). Simulation is one approach that seems to be accepted as a tool for skills training with the integration of haptic devices that provide force feedback and tactile sensors to increase realism (Panait & Akkary, 2009).

Currently, in Hospital Universiti Kebangsaan Malaysia, a dummy set box is being used as a medium for laparoscopy training (Norkhairani, Halimah, & Azlina, 2011). This method has been observed and a few weaknesses have been identified such as the absence of realism, accurate pressure and grasp level measurements (Norkhairani et al., 2011). The limitations for the trainee to be present in the actual operation room make it even worse (Norkhairani et al., 2012).

Simulation has been widely used to provide a realistic environment that will assist people in shaping their skills. It is normally applied in those areas where the real environment cannot be used as a training medium due to security and social issues. Areas such as the military, aerospace, medical and educational fields have received great attention from researchers in developing high fidelity simulations. Simulation plays an important role in the medical world since this is a risky domain. Simulation can be used to replace expensive physical mock-ups with less expensive and easily modifiable computational mock-ups (Hollerbach, 2000). Thus the need for suitable and high quality simulation is vital to provide a natural experience for surgeons before they really deal with the actual human being. Providing a high quality simulation is a complex task that must cover various activities before it can be delivered to the user (Fauziah, Aziz, & Abdul Razak, 2005).

In Malaysia, the use of haptic devices is considered to be new a field but the research in this area is growing. This research was conducted to discover how SPLasH (*Simulasi Pembedahan Laparoskopi dengan Elemen Haptik*),which integrates visual informatics components, can be implemented at the surgical department for medical students training at HUKM. The main focus of this research was to provide a simulation that could represent an almost accurate pressure and grasp level. The procedure for hernia repair was chosen as the subject for the simulation. The research was divided into three main phases:

1. The preliminary analysis of the SPLasH requirement.
2. The design and development of the SPLasH simulation.
3. The user acceptance test for SPLasH as a tool to aid laparoscopy surgery training at HUKM.

The preliminary analysis was carried out and a few major issues were identified that needed to be catered to. One of the most important findings was that the current training method did not provide an appropriate pressure and grasp experience to the trainee, where this element is a vital part of the procedure (Norkhairani et al., 2013). In the second phase, which involved the design and development of SPLasH, some difficulties were encountered since only a small number of references on technical solutions, which were not in overview mode, could be used as basic guidelines in choosing the right programming or application development. Previous work focused on haptic rendering, collision detection, issues in haptic development and a few more (Panait et al., 2009; Rose et al., 2001; Sansregret et al., 2009; Basdogan et al., 2001; Dunkin et al., 2007; Webstring-van der Putten et al., 2009; Webstring-van der Putten et al., 2010). However, a number of researchers (Lamata et al., 2007; Alan et al., 2003) did discuss technical solutions and developments, but in very light and general terms.

After an extensive review of previous works on certain parts of the designing of virtual reality applications and simulations, as well as a wide reading of a programmer's guide for one type of haptic device, this paper will discuss the technical solution frameworks that will serve as a basic and foundational paper for new and novice researchers in haptic applications to give them an overview of how this kind of application can be developed. The basic framework and class structure will be discussed with the support of previous work. This is a part of the second phase of the whole research. It is expected that this paper will give some idea to the new researcher in haptic applications, especially in Malaysia, to get started and to enrich the literature in visual informatics generally and haptics specifically.

LITERATURE REVIEW

Simulating a surgery is much more complex compared to other areas such as flight simulations (Lamata et al., 2007; Seymour & Rotnes, 2006). Surgical simulators have some unique features that make them more complex compared to simulations in other area. The great difficulty of modelling live organs, the visual update rate, the force and positions are part of the unique identity of surgical simulations. In their research, Seymour and Rotnes (2006) divided the challenges in developing medical simulations into five major areas that have been identified as:

1. Coupling of actual surgical instruments with the simulator (the surgical interface).
2. Simulation of objects (geometric and physical modelling).
3. Simulation of the interaction between the objects (object collision).
4. Display of both the operative field and the simulated operating room (e.g. patient, equipment, team members).
5. Signal processing for visualization and graphics rendering (texturing, light and smoke phenomena, fluids).

These issues are quite familiar as they have been addressed by researchers in the electronics gaming industry. In fact, the developers of some simulations have taken advantage of technological innovations in the electronics gaming industry (Lamata et al., 2007).

Surgical Simulation Components

Lamata et al. (2007) described three basic elements in surgical simulations: a haptic interface, a monitor to view the surgical scene and a computer to manage the interaction between the haptic interface/monitor and the visual models (tissues, organs, tools). For the application part, four main generic modules that are interconnected to each other to support the simulation have been developed. It can be simplified as shown in Figure 1.

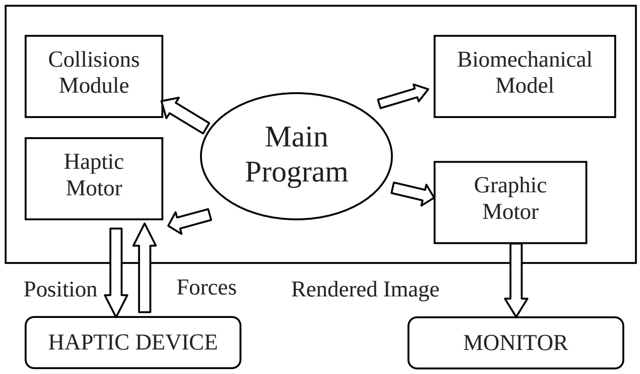

Figure 1. Four main generic modules for surgical simulations adapted from (Lamata et al., 2007).

The graphic motor renders the geometry to be visually displayed on the monitor while the haptic motor will read the position of the haptic device and return the feedback on the appropriate forces to the user. The Biomechanical Model will calculate the behaviour and deformation of the organ that appears on the virtual scene, which is simulated with a T2-Mesh mass-spring model or another alternative called ParSys (Lamata et al., 2007). The T2-Mesh model is a surface model that defines a set of nodes on the surface of the objects with a mass assigned to each. The T2-Mesh mass-spring model is an iterative model that has a risk of instabilities and oscillations. Therefore, the ParSys has been developed to overcome the limitations of the T2-Mesh. The ParSys is composed of particles where the volumetric behaviour is given and guaranteed by its structure that allows simple management of topological changes.

The Collisions Module is purposely designed to detect and handle collisions between the elements (organs/tools) and this is the part that makes surgical simulations complex. A clear distinction must be made between:

1. Detection of collisions.
2. Handling of collisions.
3. Determination of overlapping regions.
4. Solving of overlapping situations.

Alan et al. (2003) did a review of surgical simulators in terms of applications, technology and education. They divided the surgical simulator into three main types known as needle-based procedure, minimally-invasive procedure and open surgery. These differ in terms of the instruments used, incisions and types of patient's problems. These three types of surgery require a set of similar and different skills by the surgeon. As a result of a survey, they clearly defined a few components in a surgical simulator as shown in Table 1.

Table 1

Components of Surgical Simulator

Components	Subcomponents	Descriptions
Deformable models	Mass spring models	- Classified as kinematically-and physically-based - Kinematics does not consider the effects of object mass, force or other physical properties during deformations. - Physically-based model incorporates material properties, and commonly uses a mass spring model and a finite element model.
	Finite element models	
Collision detection	Collision bounding	- Handles interaction between tissues, organs and tools, and must be performed effectively. - Collision bounding normally uses a sphere or bounding box. - When intersection happens, there is a need to determine the bounding volume intersection.
	Collision refinement	- Few methods of detection and bounding setting are available.
Visual and haptic display	Tactile feedback	- Tactile feedback is sensed by the receptor that is closest to the skin, normally the fingertips. Normally used for presenting local shapes, textures and local compliances.
	Force feedback	- Force feedback deals with update rate when the user feeds the force in order to achieve high fidelity. There are a few devices available for this task from SensAble Technologies, Immersion and many more.
	Visual displays	- Visual displays are a key feature for simulation. The categories available are head-mounted, stereoscopic, monitor, environmental display and retinal display. Each has a different level of viewing.

(continued)

Components	Subcomponents	Descriptions
Tissue modelling and characterization		- Very complex and has behaviours such as: 1. Viscoelastic – stress-strain depends on the rate of deformation. 2. Inhomogeneous – varies through the tissue volume. 3. Anisotropic – varies with direction. - A key question in modelling tissue behaviour for simulation is the level of accuracy required. - Another issue to be considered is how tissue damage caused by excessive force imparted by the surgical instruments can be predicted.
Performance and training		- Goal of simulation is training in the skills and teaching the knowledge necessary to successfully perform a procedure. - Important distinction is ability versus skill. - Ability is a relatively stable capability or aptitude “that underlies (or supports) performance in a number of tasks or activities” (Schmidt & Lee, 1998). - Skill is learned or trained, and may depend on a range of underlying abilities (Patrick, 1992). - Can be determined using task analysis with various types of validity.

Technical Considerations in Haptic Application Designs

Hollerbach (2000) stated that the current haptic devices have some limitations in terms of workspace and dexterity. It is hoped that there will be gradual advancements to overcome the following issues that were reviewed:

Internal versus external forces

Most haptic interfaces are essentially force-reflecting hand controllers, where only external forces of contact are provided and not internal forces of grasping. Ideally it should provide both of them.

Workspace

Desktop haptic interfaces normally have a small workspace, so it only becomes the hand controller instead of mimicking the natural use of our arms and hands. How the size of workspace influences the task requirements is not known. It seems as though the desktop is good at probing actions in a small region, but in a real environment a larger workspace is needed and therefore a careful study needs to be carried out.

Grounded versus ungrounded haptic interfaces

While hand controllers naturally display external forces, another approach purely generates the internal forces via a portable force-reflecting master glove. What a pure internal force capability buys you compared to the external force capability is a question that has just been addressed (Richard & Cutkosky, 1997). One issue that has been raised is that an ungrounded haptic interface (i.e. a portable exoskeleton) generates reaction forces against the body as opposed to grounded haptic interfaces.

Auxiliary controls for mobility

Common haptic interfaces are not portable and have a limited workspace such that for movement to a new location the haptic device needs an auxiliary control. Common methods used are rate control and re-indexing. How natural such control functions are in simulating unrestrained mobility while manipulating remains to be seen.

The prototyping of human actions is one area that looks promising in the research into haptic devices. For example, automobile companies need to test whether their assembly staffs are able to assemble auto parts or perhaps expert surgeons may need to know whether a novice surgeon is able to carry out a procedure. Thus, there is a need for a mock session for this purpose because the test cannot be done in the actual environment. This means that human actions as well as objects need to be well prototyped. Consequently, a haptic interface to a simulation is the best way to predict whether humans are able to perform tasks comfortably and completely.

Accurate and Interactive Applications

Geometric interaction and dynamic interactions are two technical considerations that need to be catered to in developing an accurate and interactive simulation. Geometric interaction deals with computing where the object touches and it can be arbitrarily difficult depending on the geometry of the environment. Even though the computation is global in nature because anything could be touching anything else, normally the assumption is made that there is a strong locality in the geometric interactions, so only a few points need to be considered. There are two parts to the common computation strategy; do a global minimum distance computation of the isolated points of potential contact, and do a local geometric computation of those isolated points (Hollerbach, 2000). A global minimum distance computation is normally slower than a local geometric computation that is much faster to satisfy haptic requirements. The underlying geometric representation influence shows these two steps can be carried out. This leads to the issue of collision-detection research that is currently dominated by polygonal models of objects. For convex shapes, fast minimum distance calculations can be done while for more complex objects, it can be composed of convex parts. However, the issue arises as to how well sculpted objects can be represented by polygons, especially in the regions of high curvature.

This issue is overcome by employing overwhelming representations that are normally used in mechanical CAD systems, called NURBS, which stands for non-uniform rational B-spline surfaces. The advantage of NURBS is that it has additional trimming curves which include exact models and parsimony. However, many roboticists shun NURBS because of unfamiliarity. Researches by (Hollerbach, 2000; Richard & Cutkosky, 1997; Thompson & Cohen, 1999) indicate that the other issues surrounding NURBS currently (Thompson and Cohen, 1999) are:

1. Surface-to-surface NURBS computation – this computation is more complex and yet is necessary in order to fully model the object interactions.
2. Bounding volume tool chest for global minimum distance computations – various forms of bounding volumes have been proposed, including axis-aligned or oriented-bounding boxes and spheres. It is easy to come up with examples where any particular bounding choice is poor. The idea is to develop for a particular object. The cost of choosing the most suitable bounding volume has to be traded off against the efficiency of sticking to just a single type.
3. Time coherence – from instant to instant the position of the object does not change much and this notion should permit speedups in the minimum distance calculation.
4. Hybrid model interaction –NURBS is not necessarily the best choice in every circumstance. For example, the human body may be more suitably modelled by implicit surfaces and yet needs to interact with objects. The geometric method should be able to handle hybrid model interactions such as implicit surfaces versus NURBS.

Dynamic interactions with haptic applications are comprised of two parts; the contact forces between the haptic interface and the object surfaces, and the forces of interaction between objects. Later, a generic problem arose as to how to simulate the dynamics of the environment, which deals with solution requirements, not particularly about haptics. Haptic rendering is an approximate process to enable it because the solution of the dynamics of objects interacting should be as exact as possible. This is due to the issue of what feels realistic to the user. The haptic rendering process needs to be well understood because it deals with how to present normal contact forces, friction forces and texture forces.

From extensive discussions on the above section, it can be concluded that technical considerations need to be carefully studied and understood in order to provide a good simulation with a haptic device. Each of the issues that have been highlighted has an impact on the accuracy and ability of the simulations. The next section will discuss how previous work findings were taken into consideration in designing a solution framework for the SPLasH research project.

FINDINGS

It has been clearly shown in previous works that a few components for modelling, collision detection and handling together with haptic control are essential in any surgical simulation development. Also some of the reviews of the challenges and issues in haptic research need to be carefully studied and embedded into the design. But what are the programming or application alternatives that are available for each of the components? Single application solutions also need to be clearly defined to help new researchers to kick-start the simulation development. Based on the above discussion and the arising issues, an overview of the solution framework for SPLasH is discussed in the next section.

Proposed Solution Framework for SPLasH

For this research, the Phantom Omni and the Phantom Desktop have been selected as the haptic devices to be integrated into the simulation for laparoscopy surgery. Since there is limited knowledge and budget constraints, the grounded haptic interface has been chosen for this project. Another reason is that the surgical procedure does not involve the simulation of the movements of the whole body. The ungrounded haptic interface needs to be considered if the simulations need to mimic human actions that involve the movements of the whole body. So for this project, the selected interface should be enough to fulfil the needs of the project.

The surgical scene that was chosen was the procedure for the repair of a hernia. Four main components were proposed with a list of programming solutions or applications for each part of the components (SensAble Technologies Inc., 2008). The components stated here were there results of a previous work that had been reviewed. To implement each of the components, a review of the available programme was carried out and as a result four basic classes were identified for creation. A brief discussion of each of the components and the classes will be given in the next section. The framework is shown in Figure 2.

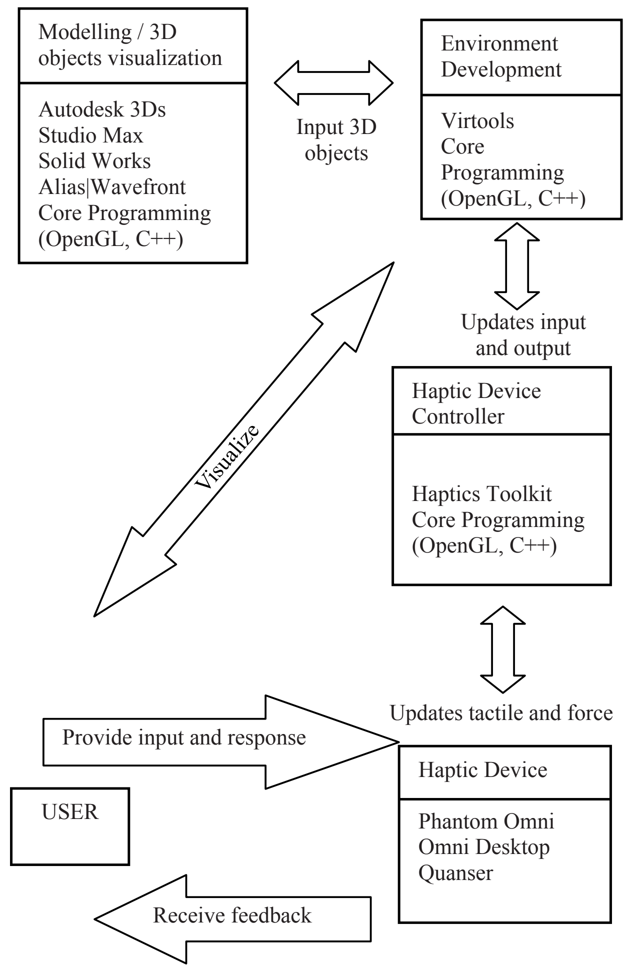

Figure 2. Proposed solution framework for SPLasH.

This framework was derived from previous work done by Lamata et al., (2007); SensAble Technologies (2008); Seymour and Rotnes, (2006); Alan et al., (2003); S. Tuchsmid et al. (2006) and technical guidelines provided by SensAble Technologies (2008) and Hsiu-Mei Huang et al. (2009). In general, it has three soft components; modelling, environment and controller, while one hard component is the haptic device. These components are basically more like what had been done in the previous work, with the addition of an available programming language or application for each of the components. The next section will discuss each of the components to provide the basic understanding to the new researcher.

Modelling

Modelling is the component that works on creating the organs/tissues, instruments/tools and view/scene. Modelling can make use of the available 3D applications on the market, for example Autodesk 3Ds Studio Max, Solid Works and Alias Wavefront. These applications can be used if the modelling tends to produce a 3D object. Adobe Flash can offer the features for 2D objects. Each of these basically has common features and ability; the familiarity factor will determine which application should be chosen. The objects produced must be saved in .obj format to enable it to be pushed into the Tri-Mesh class in a dedicated class structure which will be discussed in Section 3.2, i.e. the class structure of SPLasH. Instead of using 3D applications, core programming such as C++ and OpenGL can also be used to create the model. Even though they are relatively difficult to encode, programmers have the most control over the programme together with an efficient graphic rendering pipeline (Hsiu-Mei Huang et. al., 2009). For this research, a single programming solution has been chosen where the modelling parts (though not all parts) will be developed using the OpenGL core programming. This is because the Application Programming Interface (API) to the device used is controlled in OpenGL.

Environment

The environment refers to the surgical scene. Regardless of what type of simulation is developed, the environment plays an important role for the user. It represents the actual scene and environment so that the users will have the immersiveness during their session with the simulator. This reflects back to the geometric and dynamic interactions that were discussed previously (Richard & Cutkosky, 1997). For this research, the scenes that need to be portrayed are:

1. Abdomen surface before insertion of the laparoscope and other instruments.
2. Internal view of the abdomen with the hernia problem shown, tissues and suturing process.
3. Abdomen surface after the procedure with the small incisions on the left and right sides, and also the navel that has been sutured.

Virtools is an application that can be used to develop the environment. It is relatively simple to build and the source project is more intuitively understood as the project components are graphically based as opposed to C++ or Java or OpenGL (Hsiu-Mei Huang et al., 2009). Therefore, it requires less programming skills. Besides that, it also provides a preview function that enables programmers to view the effects or scene immediately. The other side of it is that the programmer has little control over the application where they might only use built-in components. Programmers can customize the programme but by doing so they lose the simple-to-build advantage. Again, core programming can also be used to develop an environment that offers greater flexibility.

Controller

Depending on the type of haptic device used, it normally comes with a software development kit (SDK) to control the interaction between the application and the device. As for this research, the PHANTOM device was chosen and it came with the OpenHaptics® toolkit (SensAble Technologies, 2008) that included the QuickHaptics micro application programming interface (API), a Haptic Device API (HDAPI), a Haptic Library API (HLAPI), Utilities and a PHANTOM® Device Drivers (PDD) that needed to be synchronized to achieve the desired results. QuickHaptics is a micro API that makes it fast and easy to write new haptic applications or to add haptics to existing applications. Built-in geometry parsers and intelligent default parameters make it possible to set up haptics/graphic scenes with a minimal amount of code. The HDAPI provides low-level access to the haptic device, enables haptics programmers to render forces directly, offers control over configuring the runtime behaviour of the drivers, and provides convenient utility features and debugging aids. The HLAPI provides high-level haptic rendering and is designed to be familiar to OpenGL® API programmers. It allows significant reuse of existing OpenGL codes and greatly simplifies the synchronization of the haptics and graphics threads. The PHANTOM Device Drivers support all types of PHANTOM devices. The overview of the OpenHaptic is shown in Figure 3.

The QuickHaptics micro API is implemented in C++ and defines four primary functional classes (SensAble Technologies, 2008) that are briefly discussed in the next section:

1. DeviceSpace—Workspace through which the haptic device can move.
2. QHRenderer—base class for QHWin32 and QHGLUT. An on-screen window that renders shapes from a camera viewpoint and lets the user feel those shapes with a haptic device.

3. Shape—Base class for one or more geometric objects that can be rendered both graphically and haptically.
4. Cursor—Graphical representation of the end point of the second link on the PHANTOM device. This end point is sometimes called the haptic interface point (HIP).

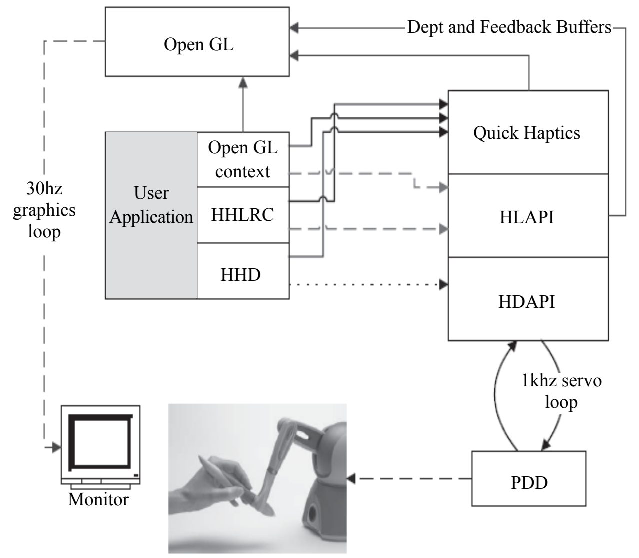

Figure 3. An overview of the OpenHaptic Toolkit (adapted from Sens Able Technologies.)

Device Space Class

Conceptually, the device space class defines the force properties and user interaction through a haptic workspace for the Phantom Omni. It manages the force effects and user callbacks. The force effects can be divided into friction, damping – degree of difficulty when moving through the space (in this case, when the user inserts a Veress needle to initiate an incision) - and also constant force. User callbacks are function calls that occur as a result of an event such as motion, haptic touch or button press.

QHWin32/QHGLUT Class

This class acts as a windowing class inherited from the QHRenderer class that defines the following:

1. Simple display list for haptics and graphics.
2. OpenGL world space to PHANTOM device space transformation.
3. Simple camera and lighting model.

World space is developed using the OpenGL (Open Graphics Library) standard that projects a three-dimensional frustum that is known as a 3-D space. It can be described and addressed by the coordinates system. Mapping between the device space and the world space is applied using a scaling factor. This may result in a small movement of the haptic device that may scale up to a much larger movement in the world space. This is very important as there is limited space for display on the screen.

Shape Class

This class defines the base class for all the geometric primitives that can be included in the world space such as line, cone, sphere, box, etc. One of the defined classes is the TriMesh class that represents a 3-D model produced by industry standard modelling programmes such as SolidWorks, 3DsMax and Lightwave, as long as the object is in STL, OBJ, 3DS and PLY formats. The deformation of the shape when touched by the haptic cursor is handled by a property.

Cursor Class

This class defines the haptic interface point. It pulls together information from all the other classes mentioned above. The device space class provides information on the location of the haptic interface point because there can be more than one haptic device and more than one cursor. For example, for this study there will be two haptic devices and two cursors that represent the device (the grasper and the needle).The World Space class provides information about the transformation that allows the device space cursor position to be drawn to the scene. For this study, this reflects the space of the internal abdominal wall where the procedure is taking place.

The shape class provides information about the objects with which the cursor will interact and how the cursor should be represented. For this study, the shape will be the abdomen (external and internal).

The cursor class default is a “blue cone”. For this study, a TriMesh model will be loaded as the cursor will be represented by the instruments that are used in the procedure (Veress needle, jaws/grasper, laparoscope and needle with thread).

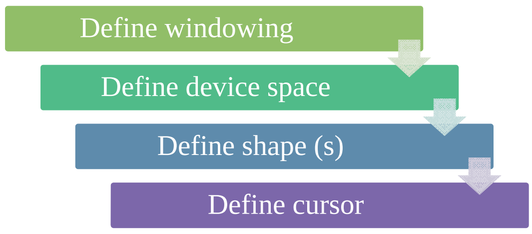

Figure 4. Flow of class defining the SPLasH using QuickHaptics.

The class structure defined in QuickHaptics is already aligned with what has been discussed in the previous work. The components remain the same, and it is only the type of application and programming language that might differ from one solution to another. This solution will be adapted in developing SPLasH, by following the flow of class that needs to be defined using OpenGL for each of the components. This design will be enhanced in a detailed description during the development and the consideration to handle two devices concurrently needs to be carefully studied to ensure it meets its purpose.

CONCLUSION

Developing surgical simulators involves multiple elements that are interconnected to each other to achieve high fidelity simulation. Thus, the determination of which technology and structure design to choose needs to be carefully studied. Technical considerations play an important role in ensuring the success of the development. Technical functionality that is not thoroughly linked to the requirements of the software can lead to developmental failure (Udechukwu Ojiako, Greenwood, & Johansen, 2005). The issues and challenges in haptic research also need to be carefully studied and understood. Technical considerations have to be given attention since they have an impact on the process of designing and developing a simulation. This framework has been designed after an extensive review of previous work and synchronization with current research. This framework can serve as a ground reading for new researchers before they start on their development. It is not limited to only surgical simulators but is also suitable for any other type of simulator that involves haptic devices.

Three main components; modelling, environment and controller, need to be focussed on during the design and development. Modelling refers to those components that represent the visualization of objects, devices or equipment taking part in the simulation. There are a few applications available in the market that can fulfil this requirement. The environment represents the scenario or scenes in the simulation domain, while the controller is a component that controls the interaction between the modelling and the environment.

Four basic classes have been discussed to give some idea to newcomers on how they can be implemented. The device space class can be utilized to manage the boundaries of the device movements in a simulated environment where the force effects will take part in providing feedback to the user. Meanwhile, the QHWin32/QHGLUT complements the device space class in mapping the area in the simulation and the real world using a scaling factor. The shape class plays the role as a holder for all the geometric primitives that can be included in the simulation. Lastly, the cursor class acts as a haptic interface point that can be represented by the TriMesh model in certain parts of the simulation.

Currently the framework is undergoing some testing on its applicability in the development phase. Some issues might arise such as the transformation of the modelling format, integration with the space and controlling the device. After the completion of the framework applicability test, the research will move forward to refine the communication between the surfaces and the engine of the simulation before a user acceptance test can be carried out. It is hoped that this paper will enrich the literature in the haptics field as well as in the visual informatics field.

ACKNOWLEDGEMENTS

This research has been funded by a grant under *Projek Arus Perdana* (UKM-AP-ICT-16-2009) entitled ‘Multi-Display Interactive Visualization Environment on Haptic Horizontal Surface (Mi VH Tabletop)’. The authors would like to thank the Visual Informatics (VI) group of The Institute of Visual Informatics, UKM and especially Prof. Dato’ Dr. Halimah Badioze Zaman, Associate Prof. Dr. Azlina Ahmad and Dr. Dayang Anita Abdul Aziz (HUKM) for supervising the research.

REFERENCES

Alan Liu, F., Tendick, K., Clerly, & Kaufmann C., (2003). A survey of surgical simulation: Application, technology and education. *Presence, 12*(6), 599–614.

Basdogan, C., Chih-Hao Ho, & Mandayam, A. Srinivasan. (2001). Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration. *Mechatronics, IEEE/ ASME Transactions, 6*(3), 269–285.

Dunkin, B. A., Andrales, G. L., Apelgren, K., & Mellinger, J. D. (2007). Surgical simulation: A current review. *Surgical Endoscopy*, *21*(3), 357−366.

Fauziah Baharoom, Aziz Deraman & Abdul Razak Hamdan. (2005). A survey on the current practices of software development process in Malaysia. *Journal of Information and Communication Technology*, *4*, 57–76.

Hamdorf, J. M., & Hall, J. C. (2000). Acquiring surgical skills. *British Journal of Surgery,87*(1), 28–37.

Hollerbach, M. J. (2000). *Some current issues in haptics research*. Proceeding of IEEE International Conference on Robotics and Automation, 522−525.

Hsiu-Mei Huang, et al. (2009). Developing a virtual reality learning environment for medical education. In T. Bastiaens et al., *Proceeding of World Conference on e-Learning in Corporate, Government, Healthcare and Higher Institution 2009* (pp. 1320–1329). Chesapeake, VA: AACE.

Lamata, P., Gomez, E. J., Sanchez-Margallo, F. M., Lopez, O., Monserrat, C., Garcia, V., Alberola, C., Florida, M. A. R., Ruiz, J., & Uson, J. (2007). SINERGIA laparoscopic virtual reality simulator: Didactic design and technical development. *Computer Methods and Programs in Biomedicine*, 85(3), 273–283.

Najmaldin, A. (2007). Skills training in paediatric minimal access surgery. *Journal of Paediatric Surgery*, *42*(2), 284–289.

Norkhairani Abdul Rawi, Halimah Badioze Zaman & Azlina Ahmad. (2013). Design of class components for laparoscopy surgery simulation with haptics elements: A case study of SPLasH. *Lecture Notes in Software Engineering*, *1*, (3), pp. 323–328.

Norkhairani Abdul Rawi**,** Halimah Badioze Zaman, & Azlina Ahmad. (2012). Technical solution framework for simulation with haptics elements: A case study of SPLasH (*Simulasi Pembedahan Laparoskopi dengan Elemen Haptik*). In Halimah Badioze Zaman et al., *VIIS'12*: *Visual Informatics International Seminar* (pp. 256–265). Kuala Lumpur: DSS Publishing Enterprise.

Norkhairani Abdul Rawi, Halimah Badioze Zaman & Azlina Ahmad. (2011). Simulations for laparoscopy surgery with element for medical students in HUKM: A preliminary analysis. In Halimah Badioze Zaman et al., *Visual informatics: Sustaining research and innovations - Lecture Notes in Computer Science 7066* (pp. 125−138). Selangor, Malaysia: Springer Verlag.

Panait, L., Akkary, E Bell, R. L., & Robert, K. E. (2009). The role of haptic feedback in laparoscopic simulation training. *Journal of Surgical Research,156*(2), 312–316.

Patrick, J. (1992). *Training: Research and practice.* London: Academic Press.

Richard, C., & Cutkosky, M. R. (1997*). Contact force perception with an ungrounded haptic interface*. In Proc. ASME Dynamic Systems and Control Division. ASME Int. Mech. Eng. Congress and Exhibition, Dallas, 181–187.

Rosen, J., Hannaford, B., Richards, C. G., & Sinanan, M. N. (2001). Markov modelling of minimally invasive surgery based on tool/tissue interaction and force/torque signatures for evaluating surgical skills. *IEEE Transactions on Biomedical Engineering*, *48*(5), 579−591.

Tuchschmit, S., Grassi, M., Bachofen, D., Früh, P., Thaler, M., Sze'kely. G., & Harder, M. (2006). A flexible framework for highly-modular surgical simulation systems. In M., Harders, & G., Sze'kely (Eds.), *ISBMS*, *Lecture Notes in Computer Science*, *4072* (pp. 84–92). Springer-Verlag Berlin Heidelberg.

Sansregret, A., Fried, G. M., Hasson, H., Klassen, D., Lagace', M., Gagnon, R., & Charlin, B. (2009). Choosing the right physical laparoscopic simulator? Comparison of LTS2000-ISM60 with MISTELS: Validation, correlation, and user satisfaction. *The American Journal of Surgery*, *197*(2), 258–265.

Schmidt, R. A., & Lee, T. D. (1998). *Motor control and learning: A behavioural emphasis* (3rd ed.). Champaign, IL: Human Kinetics.

SensAble Technologies. (2008). *OpenHaptics® Toolkit Programmer's Guide* Retrieved from http://www.dsc.sensable.com

Seymour, N. E., & Rotnes, J. S. (2006). Challenges to the development of complex virtual surgical simulations. *Surgical Endoscopy*, *20*, 1774–1777.

Thompson II, T., & Cohen, E. (1999). *Direct haptic rendering of complex trimmed NURBS models.* Proc. ASME Dynamic Systems and Control Div.

Udechukwu Ojiako, G., Greenwood, D. J., & Eric Johansen, D. (2005). *Journal of Information and Communication Technology*, *4*, 17−36.

Westerbring-van der Putten, E. P., Goosens, R. H., Jakimowicz, J. J. (2008). Haptics in minimally invasive surgery – A review. *Minimally Invasive Therapy and Allied Technologies*, *17* (1), 3−16.

Westerbring-van der Putten, E. P., van den Dobbelsteen, J. J. Goosens, R. H., & Dankelman, J. (2009). Force feedback requirements for efficient laparoscopic grasp control. *Ergonomics*, *52*(9), 1055–1066.

Westerbring-van der Putten, E. P., van den Dobbelsteen, J. J. Goosens, R. H., Dankelman, J. (2010). The effect of augmented feedback on grasp force in laparoscopic grasp control. *IEEE Transactions On Haptics*, *3*(4), 280–291.

Wong, J. A., & Matsumoto, E. D. (2008). Primer: Cognitive motor learning for teaching surgical skill—how are surgical skills taught and assessed? *Nature Clinical Practice Urology, 5*(1), 47–54.